



POZNAŃ UNIVERSITY  
OF ECONOMICS

# Fungible devices and ubiquitous apps

JACEK CHMIELEWSKI  
chmielewski@kti.ue.poznan.pl

Department of Information Technology  
<http://www.kti.ue.poznan.pl/>

# Outline

Fungible devices

Ubiquitous applications

Existing solutions

... and why they don't work

New approach

Device-Independent Architecture

**New possibilities!**

# The ultimate goal

Provide the intended functionality  
in the best possible way  
**on any end-device**



POZNAŃ UNIVERSITY  
OF ECONOMICS

# FUNGIBLE DEVICES

# Fungibility

**fun•gi•ble** /'fʌŋ dʒə bəl/ *adj.*

(of goods) exchangeable or replaceable,  
in whole or in part,  
for another of like nature or kind.

## fungible device

a device that can be replaced by any other device  
without hindering the provided functionality  
without the need for multiple variants of an app

# Obviously not fungible devices

in-vehicle infotainment unit != laptop

smartphone != smart TV

- Different purpose
  - Various sets of functions needed on such devices
- Different equipment and capabilities

# Almost fungible devices

smartphone  $\sim$  tablet

tablet  $\sim$  laptop

laptop  $\sim$  desktop PC

- Similar purpose
  - But different interaction contexts
- Different capabilities and interaction modes
  - UI adaptation required

# Fungible devices ?

tablet ?= tablet

smartphone ?= smartphone

- Hardware differences
  - Display: size, aspect ratio, resolution, pixel density
  - Availability of sensors and/or hardware buttons
- Software differences
  - Operating systems
  - Operating system versions (different API)



# Fungible devices

- Hardware uniformity
  - Same equipment
  - Similar parameters
    - Screen aspect ratio
    - Processing resources
- Software uniformity
  - Same OS & API

# Fungible devices



iPad mini  $\approx$  iPhone 5



Nexus 7" = Nexus 7"

# Fungible devices

In the real world device  
**fungibility is unattainable**

# There are no fungible devices

So how to  
provide the intended functionality  
in the best possible way  
**on any end-device?**



POZNAŃ UNIVERSITY  
OF ECONOMICS

# UBIQUITOUS APPLICATIONS

# Ubiquity

**u•biq•ui•tous** /yu'bɪk wɪ təs/ *adj.*

existing or being everywhere,  
esp. at the same time; omnipresent.

## **ubiquitous application**

an application that is always available, always at hand  
**no matter what device or devices a user is using**  
synchronized across all user's devices

# Ubiquitous applications

How to build  
**ubiquitous applications**  
in the world of  
**non-fungible devices**  
?

# Existing solutions

- Multiple native apps
- Cross-platform apps
  - Cross-compilers
  - Common runtime
- Web apps



# Multiple native apps

- Good performance
  - Closer to hardware
  - Code optimization possible
- Access to device features
  - Hardware features
  - OS integration (background tasks, lock screen...)
  - Connectivity (Bluetooth, accessories)
- Application markets
  - Facilitates app distribution and monetization

# Multiple native apps

- Multiple implementations required
  - Different developer skills
  - Different UI design patterns
  - Result
    - high development and maintenance costs
- OS inconsistencies
  - Hard to recreate some functions on every OS

# Cross-platform apps

- Consistent API across devices and OSs
- Only single implementation required
- Single toolkit and developer skillset
  - Decreases costs and development time
- Possibility to use application markets

# Cross-platform apps

- Usually worse performance
  - Inefficient code (translated or mediated)
- Restrictive frameworks and tools
  - Support for a specific programming language
  - Fixed project structure
  - Limited UI capabilities
- Restricted API in comparison to OS
- Delayed support for newest features

# Web apps

- Very popular platform
  - Runtime is available on every device
- Specific variant of cross-platform approach
  - Common runtime – web browser
  - Single language set – HTML, CSS, JavaScript
  - Fairly consistent but even more restricted API
- Performance issues
- No proper applications marketplace

# Existing solutions

- Multiple native apps
  - Best performance and access to device features
  - Highest development and maintenance costs
    - Caused by multiple implementations
- Cross-platform apps
  - Better costs and development speed
  - Worse performance and platform restrictions
    - Platform lock-in

# Ubiquitous applications

How to

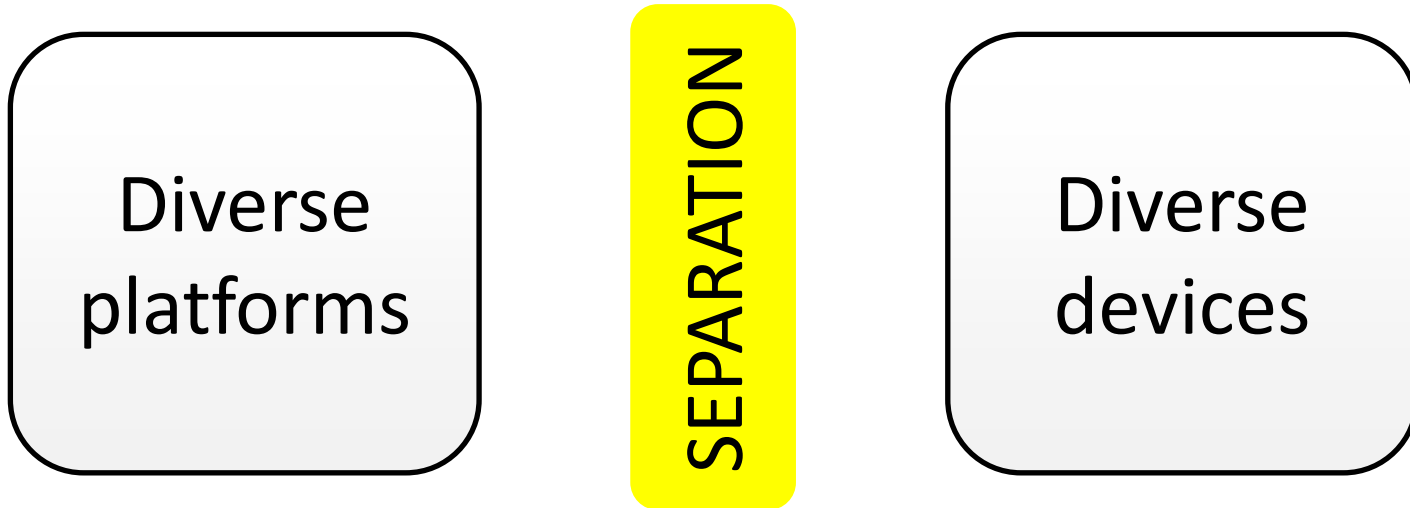
**implement once**

but avoid

**platform restrictions**

?

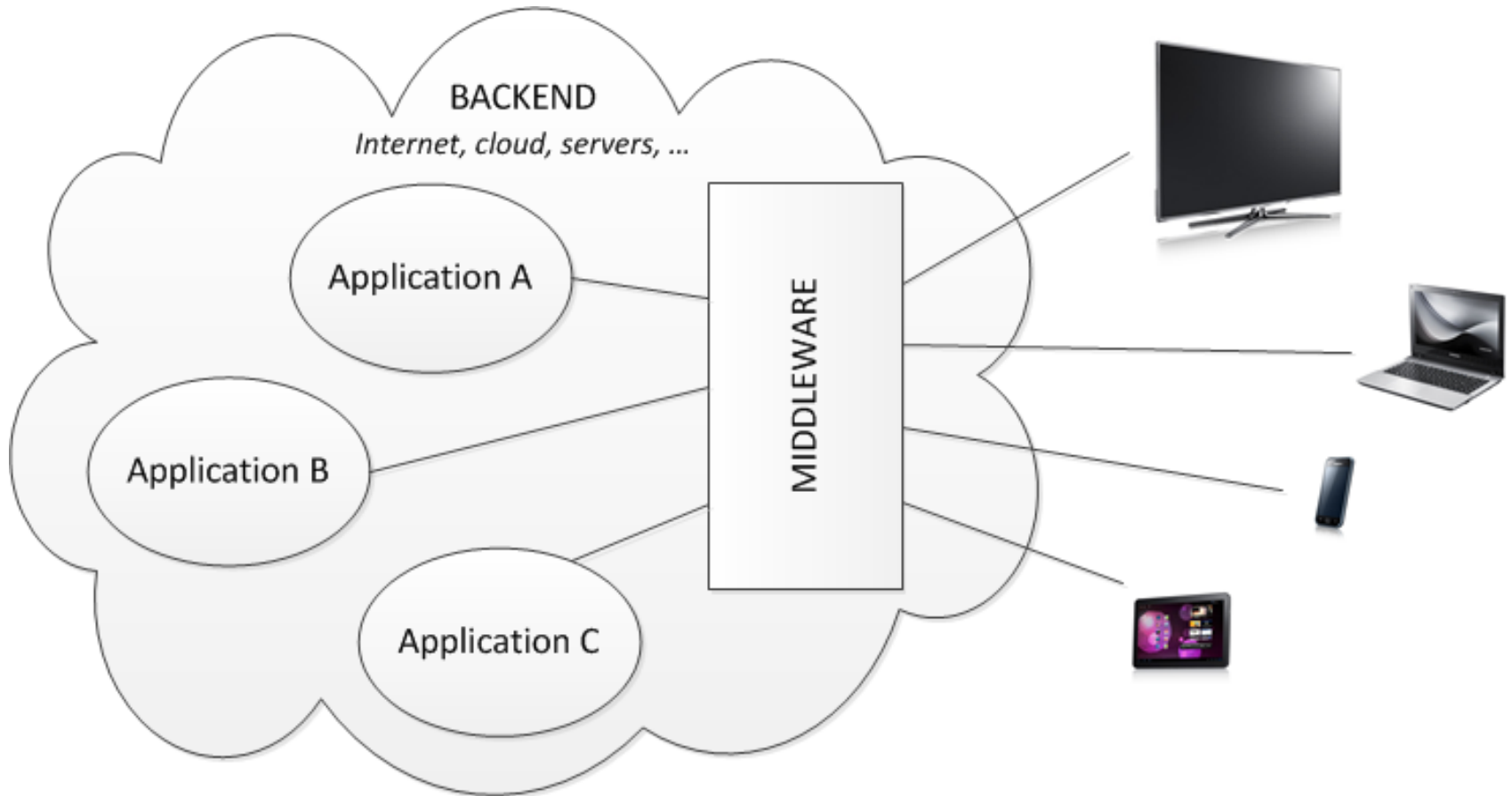
# New approach



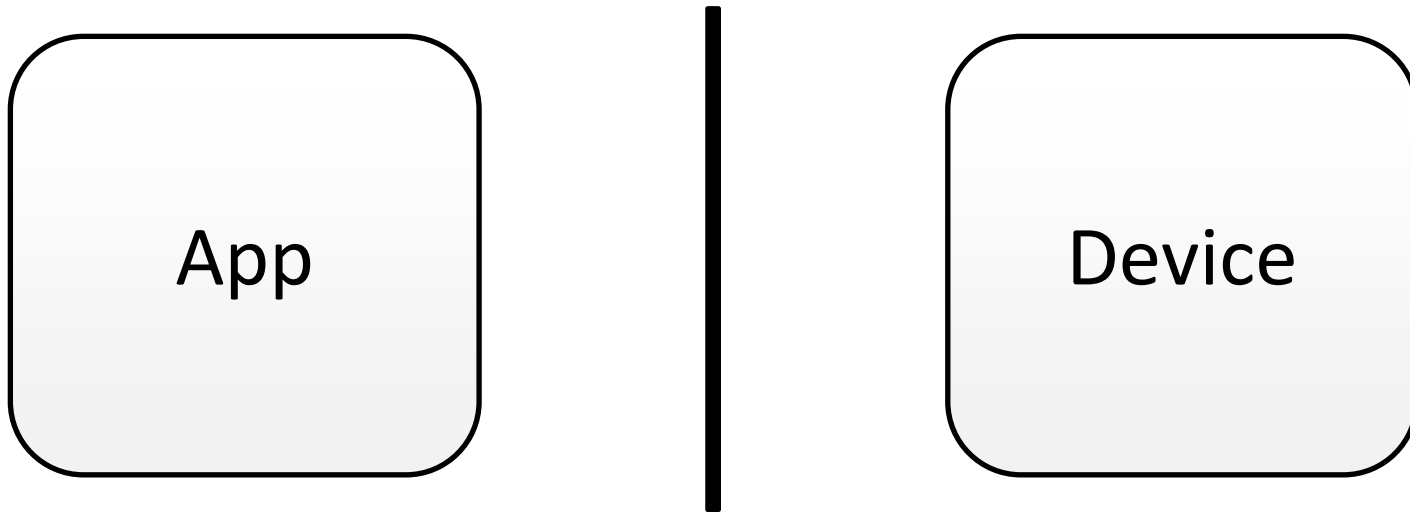
- Universal middleware layer
- Standard protocol



# Separation concept

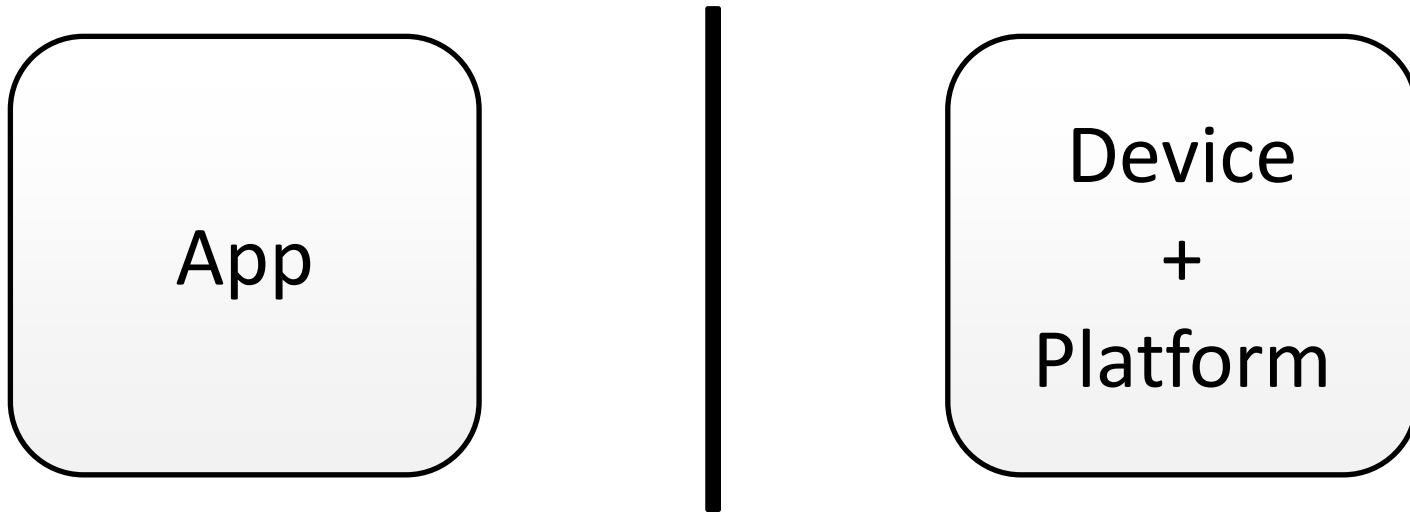


# Separation



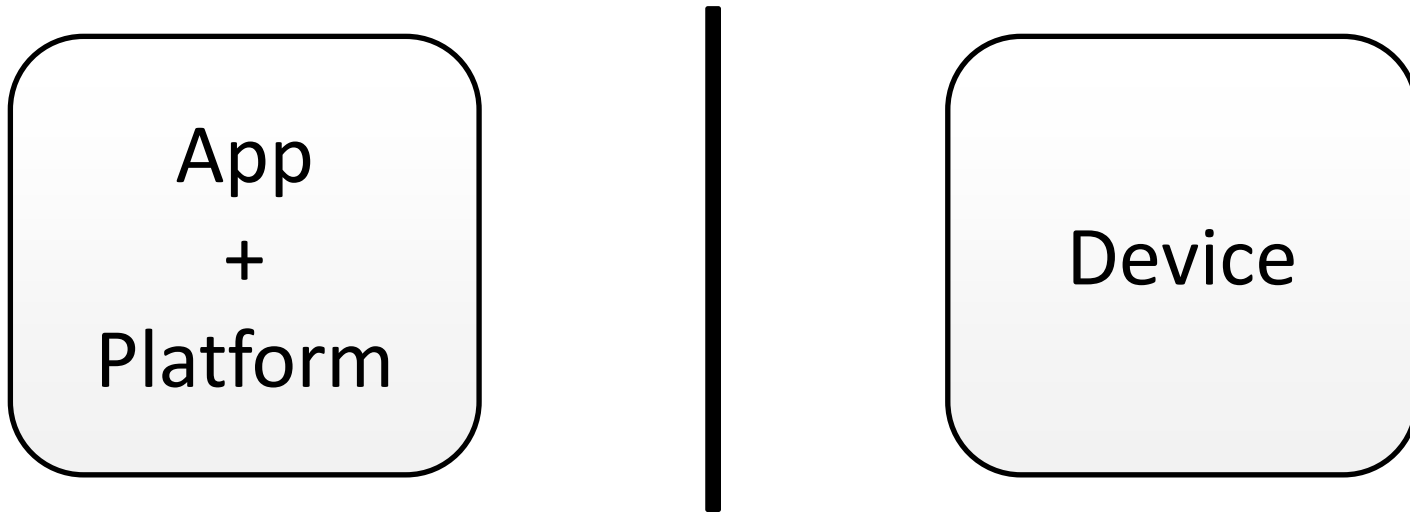
**Where to draw the line?**

# Separation: devices – apps



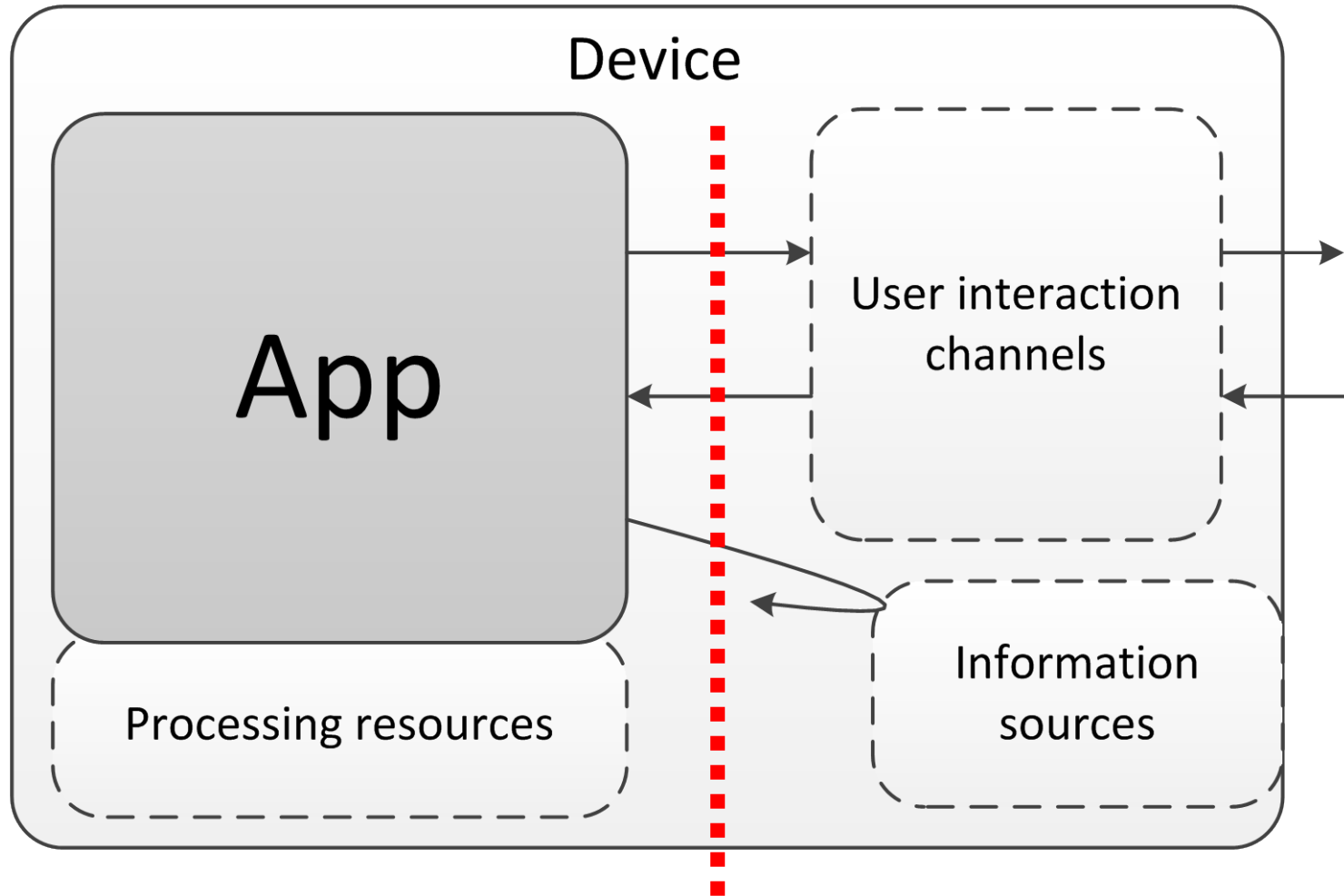
Common approach is to  
separate at the API level

# Separation: devices – apps

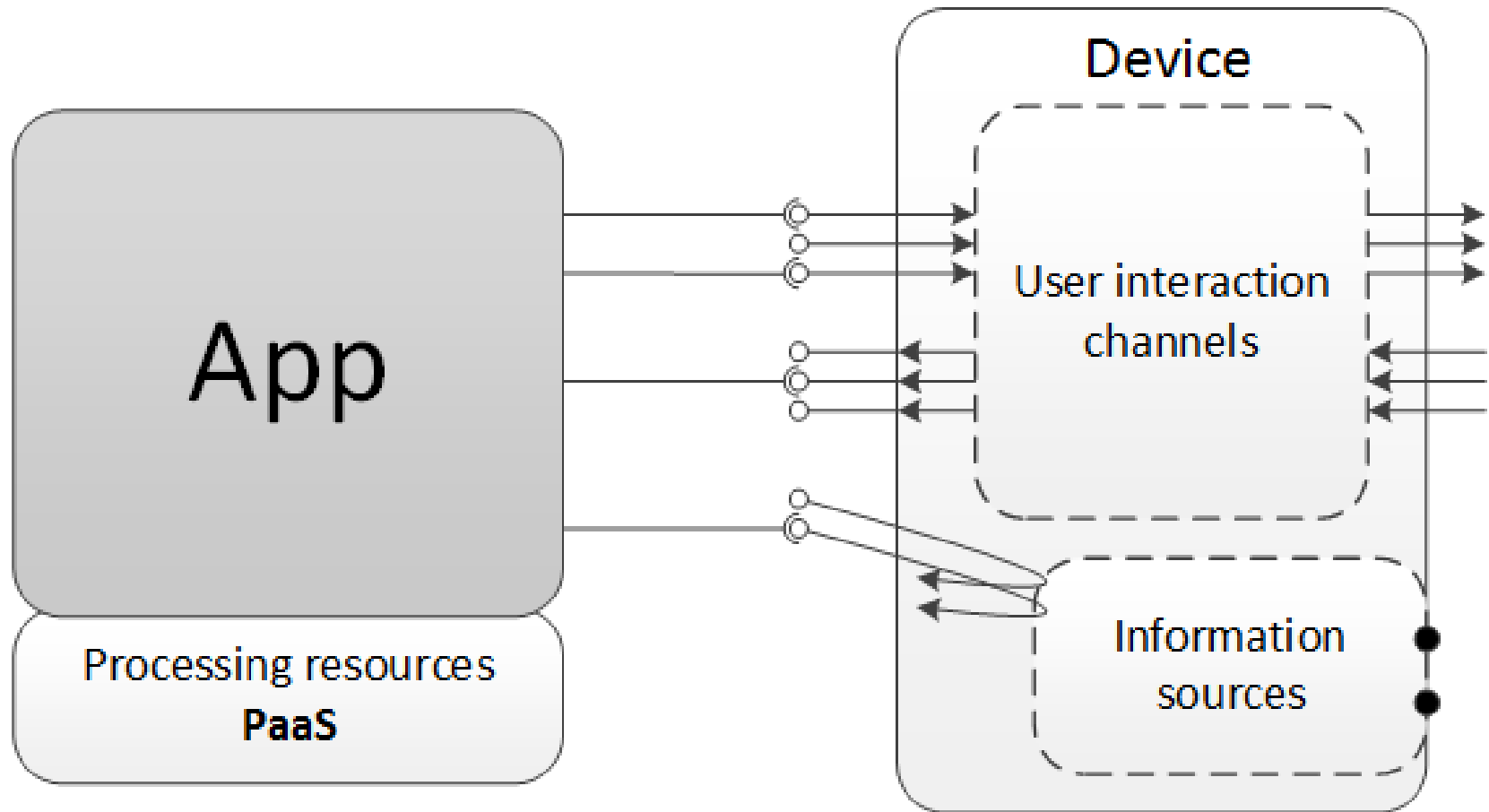


We need to  
**abstract a device**  
not a platform

# Device decomposition



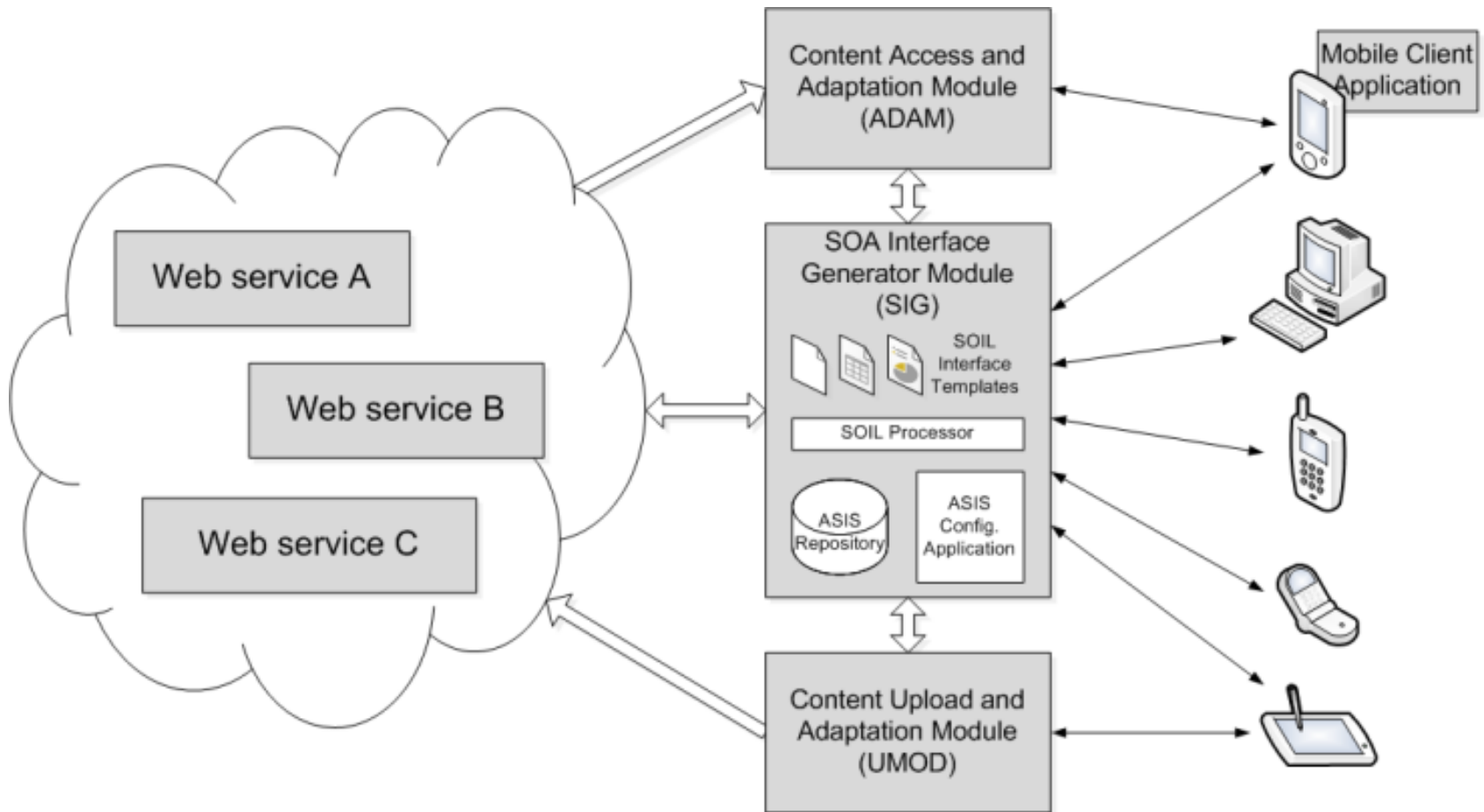
# Device-Independent Architecture



# Device-Independent Architecture

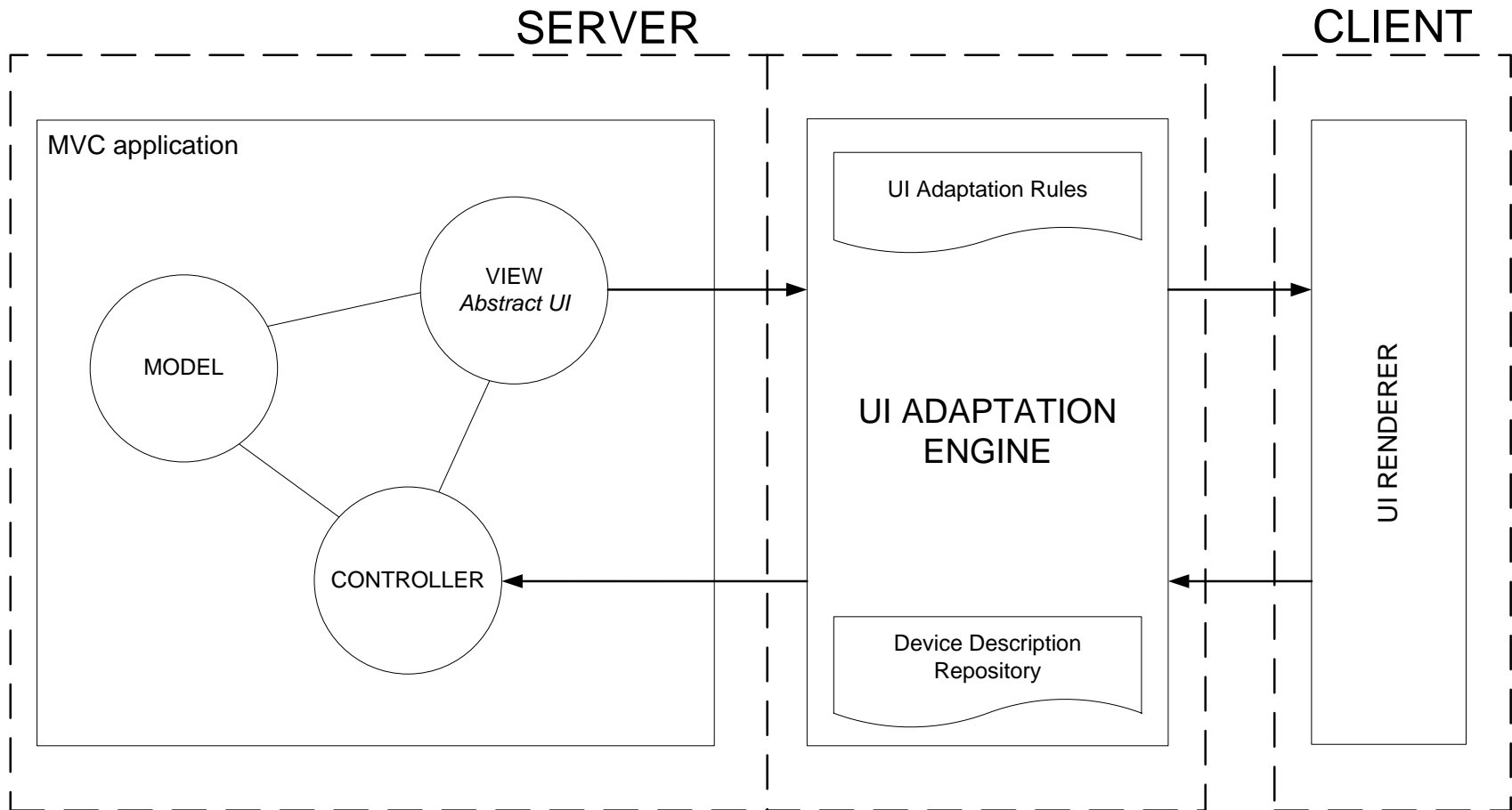
- Application executed outside of a device
  - Only single implementation is necessary
  - Any software platform is allowed
- Universal access to device sensors
  - Sensors can be queried as any external service
- User Interface abstraction
  - Multiple UI generation patterns are possible

# DIA example – IT-SOA

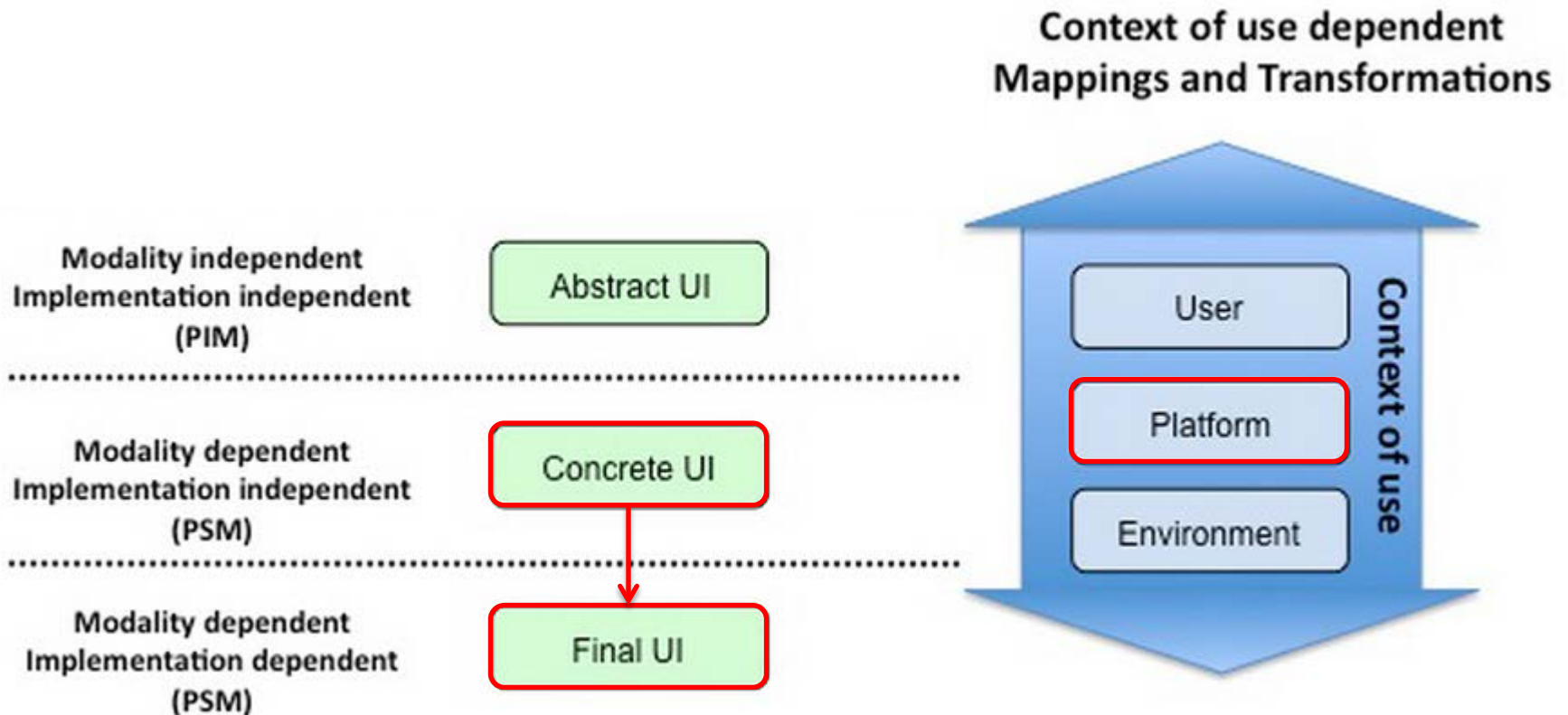




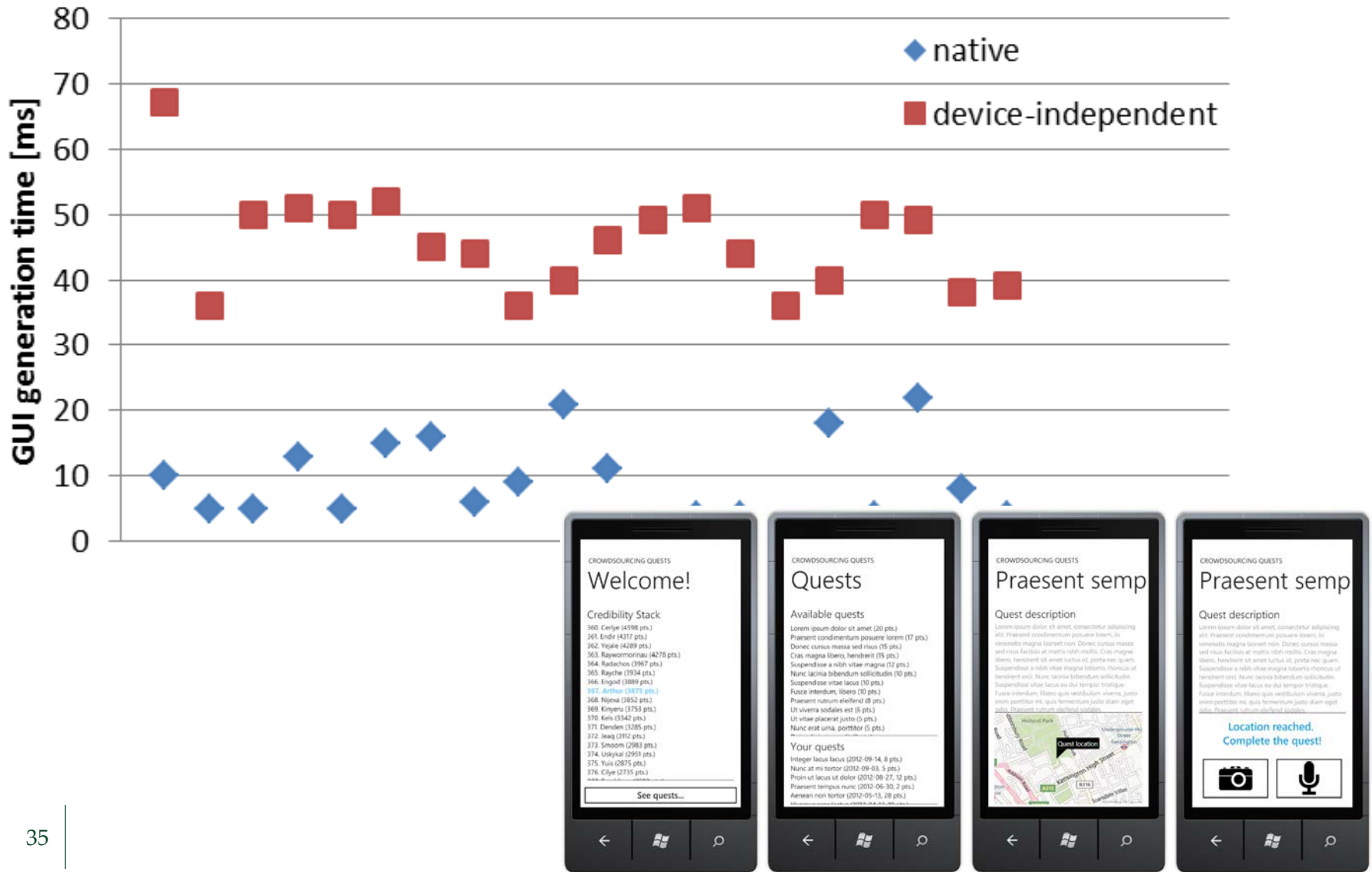
# DIA example – UbiqUI



# DIA example 3 – uBankApps



# DIA experiment – UI generation



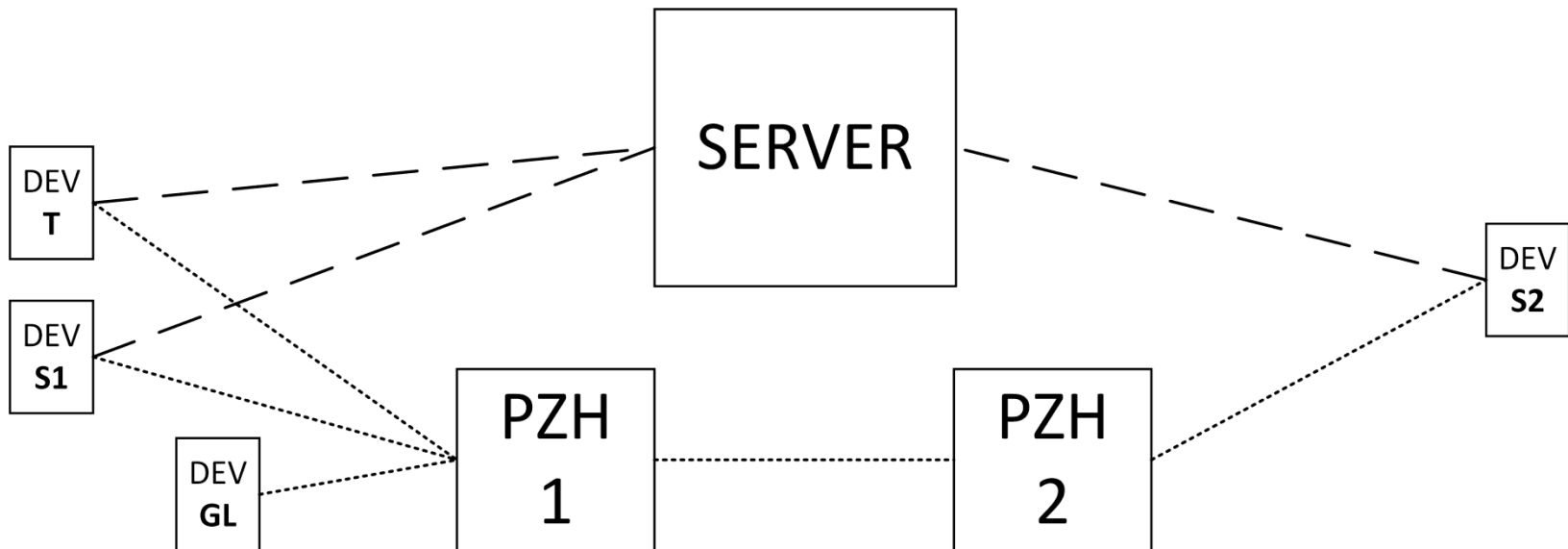
# Other benefits

- Unlimited performance
  - Not constrained by device capabilities
- Potentially better battery life
  - No battery-draining processing on a device

# New possibilities

- Applications can follow users
  - Users can switch from one device to another
  - It is possible to seamlessly migrate app UI state
- Multi-device scenarios
  - Users may use multiple devices for the same app
  - Device detection, authorization and communication solutions necessary

# DIA experiment – multi-device



# DIA experiment – multi-device



Thursday, 14 February 2013, 10:44

“ Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed vel sem vitae erat sagittis ullamcorper eu imperdiet elit. Nunc pretium dictum congue.  
@ Ratajczaka 27, 61-815 Poznań, Poland

10:44

“ Aenean ac mauris eget leo egestas hendrenit. In hac habitasse platea dictumst. Fusce non mi libero. Nulla bibendum neque ac nibh lobortis eu congue nunc hendrenit. Nulla arcu nulla, pretium id interdum in, cursus eleifend dolor. Aliquam consectetur euismod ultricies.  
@ Ratajczaka 27, 61-815 Poznań, Poland

Friday, 29 June 2012, 16:13



@ Schiffbauerdamm 25, 10117 Berlin, Germany





POZNAŃ UNIVERSITY  
OF ECONOMICS

# Thank You!

Fungible devices and ubiquitous apps

JACEK CHMIELEWSKI  
chmielewski@kti.ue.poznan.pl

Department of Information Technology  
<http://www.kti.ue.poznan.pl/>



**Solutions and tools  
for cross-platform  
and multi-device  
applications**



**Jacek Chmielewski**  
jacek.chmielewski@ubliko.com

[www.ubliko.com](http://www.ubliko.com)  
[office@ubliko.com](mailto:office@ubliko.com)

**ubliko**